AI DRIVEN HAND GESTURE RECOGNITION

Mr.R,Raja Subramanian
Computer Science and Engineering
Kalasalingam Academy of Research and Education
rajasubramanian.r@klu.ac.in

Marietty Sai Murali
Computer Science and Engineering
Kalasalingam Academy of Research and Education
m.saimuraliroyalram@gmail.com

Yalla Sireesha
Computer Science and Engineering
Kalasalingam Academy of Research and Education
siriyalla555@gmail.com

Vedala Sai Seetha Ramakrishnamacharyulu
Computer Science and Engineering
Kalasalingam Academy of Research and Education
vedalaramakrishna4@gmail.com

Yalla Satya Praveen Kumar Reddy
Computer Science and Engineering
Kalasalingam Academy of Research and Education
yspraveen999@gmail.com

***Abstract*—Human-computer interface (HCI) is the study of how humans and computers interact. Hand gestures are a great way to communicate with people who don't understand what we're saying, especially when they don't understand what we're saying. It is also an important part of human-computer interaction. Understanding hand gestures is essential to make sure the listener understands what the speaker is trying to say or the machine understands what we are saying. The main idea of this project is to try different approaches to hand gesture recognition. In this project we work first with radar data and then with camera sensor to achieve hand gesture recognition. First, we tried to build hand gesture recognition using radar data, and since most people don't know sign language and very few interpreters, we developed an approach to real-time approach for American Sign Language based on neural network finger spelling followed by another model with MediaPipe. We propose a complex neural network (CNN) method to detect hand gestures of human behavior from camera-recorded images. The aim is to recognize hand gestures of human activities from camera images. Skin models, hand position, and orientation were applied to get the training and test data of the CNN. The hand first goes through the filter, and after applying the filter, the hand goes through a classifier that predicts the type of hand gesture. The hand position is intended to transform and rotate the image of the hand in a neutral position. Then train the CNN with the corrected image. We used machine learning, deep learning, and computer vision to create this model. Our MediaPipe model works well to detect different gestures.**

Keywords— Deep Learning, CNN, Machine Learning, Radar, Media Pipe, HCI, RNN, Sign Language.

I. INTRODUCTION

Human-computer interface (HCI) is the study of how humans and computers interact. Hand gestures are a great way to communicate with people who don't understand what we're saying, especially when they don't understand what we're saying. It is also an important part of human-computer interaction. Understanding hand gestures is essential to make sure the listener understands what the speaker is trying to say or the machine understands what we are saying. The main idea of this project is to try different approaches to hand gesture recognition. In this project, we work with radar data first and then with camera sensor to perform hand gesture recognition. First, we tried to build hand gesture recognition using radar data, and since most people don't know sign language and very few interpreters, we developed an approach in real time for American Sign Language based on neural network finger spelling, then another model with MediaPipe. We propose a complex neural network (CNN) method to detect hand gestures of human behavior from camera-recorded images. The aim is to recognize hand gestures of human activities from camera images. Skin model, hand position and orientation were applied to get training and test data from CNN. The hand first goes through the filter, and after the filter is applied, the hand goes through a classifier that predicts the type of hand gesture. The hand position is intended to transform and rotate the image of the hand in a neutral position. Then train the CNN with the corrected image. We used machine learning, deep learning, and computer vision to create this model. Our MediaPipe model works well to detect different gestures.

Because people with speech disabilities cannot communicate with their hearing and words, they must rely on sign language. All people with speech disabilities use sign language, but they find it difficult to interact with people who do not know signs (people who do not know sign language). For people who are deaf or hard of hearing, a sign language interpreter is essential. Their casual and informal communication is impeded. With recent advances in deep learning, there have been positive advances in the field of gesture recognition and motion recognition.

The proposed method attempts to translate hand movements to the English text equivalent in real time. This method uses video to record hand movements and convert them into text that can be understood by non-signers. Similar studies have been done before, focusing mainly on translating the signs of the English alphabet or relying solely on numbers. The CNN algorithm will be used to classify the hand movements. The communication gap between signatories and non-signers will be bridged through this technology. This will facilitate communication for the visually impaired.

American Sign Language is the dominant sign language Since DandM's only disability is the ability to communicate and they cannot use spoken language, the only way to communicate is through sign language. Communication is the process of exchanging thoughts and messages in different ways such as words, signals, behaviors and images. Deaf people (DandM) use their hands to make different gestures to express their ideas to others. Gestures are non-verbal messages that are exchanged and gestures are intuitively understood. This non-verbal communication is called sign language.

Sign language is a visual language and consists of 3 main components:

TABLE 1 COMPONENTS OF VISUAL SIGN LANGUAGE

Finger Spelling	Word Level Vocabulary	Non-Manual Features
Used to spell words letter by letter.	Used for the majority of communication.	Facial expressions and tongue, mouth and body.

After recognizing the gesture our camera method does even translates the sign language predicted over there.

In our first Radar Approach the gestures we try to recognize are shown in the image below:

Fig.1. Deep Soli Gesture Data

In one of our approaches, we basically focus on recognizing finger-based hand gestures and creating a pattern that can be combined to form a complete word. The gesture you want to train is shown in the picture below:

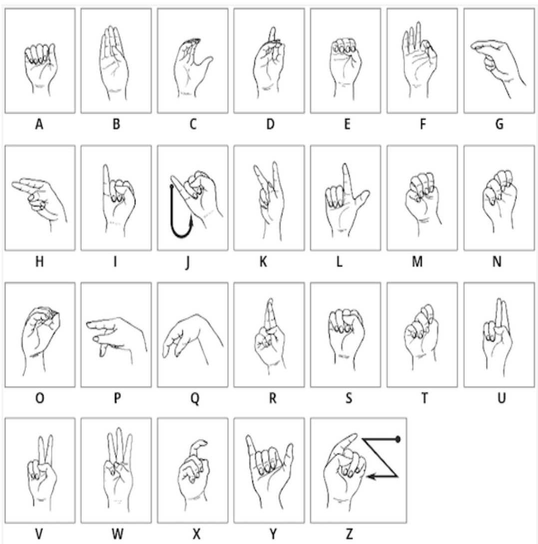

Fig.2. Sign conventions of alphabets in ASL

In our third approach with MediaPipe we'll try to classify the gestures like (i) Okay (ii) Peace (iii) Thumbs up (iv) Thumbs down (v) Call me (vi) Stop (vii) Rock (viii) Live Long (ix) Fist (x) Smile.

II. Background And Related Works

In [1], their approach is a frequency modulated continuous wave (FMCW) radar that can detect the frequency change between emitted and received electromagnetic waves according to the Doppler effect. The detection system is mounted on a radar array consisting of three continuous wave radars operating at 2.125 GHz. The decision tree algorithm is built and evaluated as a classifier for the test. Thus, gesture movements are recorded according to the Doppler effect of the frequency spectrum of the radar signal. This system has outstanding in-plane motion detection performance against bending and stretching, and achieves high detection accuracy of 92% or more.

[2] Continuous wave model with ultrasonic frequency modulation (FMCW) and ConvLSTM. One transmitter and three receivers are installed in space in different orientations. The FMCW signal emitted by the transmitter is manually reflected and then detected by the receiver. We then obtain a range Doppler map (RDM) of the received signal by processing the 2D fast Fourier transform. High resolution at 0.005 m distance and 0.03 m/s speed for hand gestures and 85.7° accuracy achieved with small size of 50 finger gesture training samples.

In [3], we use DCNN and VGG16 algorithms. ASL hand gesture movements are recorded as microwave Doppler signals using an Xband microwave Doppler radar transceiver. These hand gestures were analyzed using MATLAB. The DCNN algorithm is used to train the ASL gestures described in the spectrograms. The average validation accuracy of DCNN and VGG16 algorithms is 87.5% and 95%, respectively.

In [4], we use FMCW radar and deep neural network (DCNN). The FMCW radar operates in the 2 GHz ISM band, the effective isotropic radiated power level is 0 dBm, and the FMCW radar receives only one channel. Gesture recognition is performed using a complex deep neural network trained and tested in Micro Doppler spectroscopy. After training and validation, both methods yield 99% classification accuracy in the test set.

In [5], we worked on FMCW radar, signal processing and detection using a complex neural network (CNN). A sequence of information about the object's distance, speed, and azimuth is merged into a single input and sent to a complex neural network to learn spatial and temporal patterns. VGG10 converges to epoch 10 with verification accuracy of 92%. ResNet20 is better than VGG10 with 98% verification accuracy. The CNN LSTM has the lowest LEFT/RIGHT accuracy due to the lack of AngleOfArrival information, and the model achieves an average of 98% accuracy in the test set.

FasterRCNN, FMCW radar is used in [6]. A region based deep complex neural network (RDCNN) is proposed to detect and classify gestures measured by a frequency modulated continuous wave radar system. In addition to the μD signature, we combine the phase difference information of the signal received from the L-shaped antenna array. The input of the proposed array contains three channels, i.e. one spectrogram and two out of phase channels. . Results reached 95% (96%) Average PPV (APR) for nine gestures. Dataset

DopNet, hand gesture radar dataset used for gesture classification. The micro doppler signature is used as input to the model in [7]. Separate cumulative neural networks are used here. To accumulate without overload, the paper uses an analytic convolutional neural network model that performs deep convolution followed by point convolution. Achieved an accuracy of 9.56 on Dop Net data. Furthermore, computation time is minimized by using separable aggregates.

The Micro Doppler image dataset contains 15 types of sign language actions recorded by radar echo, as measured by the MDHandNet model in [8] The proposed MDHandNet model for sign language recognition by hand or gesture. The obtained accuracy is 97.1%. Compared with other methods, the proposed model has good performance, fewer parameters and lower computational complexity.

Radar system, Soli dataset, Dop Net dataset using Spiking neural network have been deployed in [9]. The signal to collision conversion scheme is used to encode the Doppler radar map into spike trains fed to spike neural networks. The SNN's reader signal is fed into different classifiers.

The 20BNjester dataset (OpenSource Video Dataset) is used in combination with 3DCNN and LSTM (deep learning) networks in [10]. The proposed architecture extracts spatial-temporal information from input video sequences while avoiding many computations. 3DCNN is used to extract spatial and spectral features, which are then passed to the LSTM for classification.

The HBA System Image Dataset with Infrared Hand Gestures contains 3825 images in [11]. Convolutional Neural Network for Double Teacher Mode, a non-verbal approach to extract features from infrared images of hand gestures. A robust nonlinear neural network is built with three cumulative layers. From the comparison experiments, the results demonstrate that the developed method can estimate hand gestures with a final accuracy of more than 90%.

In [14] localization and mapping algorithms are used. Images were first extracted from video streams recorded at different distances and locations. Each session consists of 300 images of 6 different layers, each layer has 50 images. It describes the extraction of hand features by extracting the number of objects and the orientation in the object of interest. After doing the experiments, we can see that the overall accuracy of the system is 95%, which is a very good result. Here, good threshold values play an important role for gesture recognition.

Many researchers have proposed numerous methods for Hand Gesture Recognition Systems.

A. Electronic Based

- Use of electronic hardware
- Complex to use
- Physical attachment with system required
- Lot of noise in transition

B. Glove Based

- Requires use of hand gloves.
- Complex to use.
- Variable glove size for more users.
- Environment Dependent.

C. Marker Based

- Use of Color markers on finger or wrist.
- Marker positions are fixed.
- Complex to use multiple markers.

D. Issues to addressed

- Latency

  Image processing can slow down significantly, creating unacceptable lag for video games and similar applications.

- Robustness

  Many gesture recognition systems do not read gestures correctly or optimally due to factors such as insufficient background light, high ambient noise, etc.

- Lack of Gesture Language

  Different users gesture differently, making it difficult to identify movements.

- Performance

The image processing involved in gesture recognition is quite resource-intensive, and applications can be difficult to run on resource-constrained devices.

III. Proposed Algorithm And Design Architecture

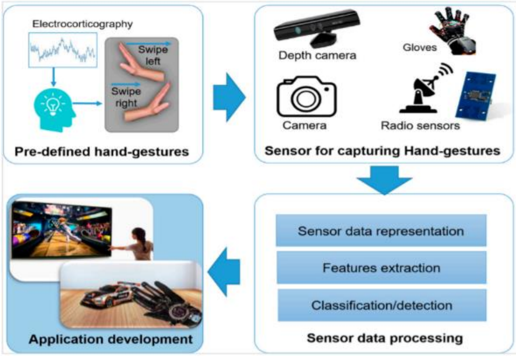

Fig.3. Hand gesture recognition architecture

Feature extraction and representation: Represent the image as a 3D matrix with dimensions such as image height and width, and each pixel's value as depth (1 for grayscale, 3 for RGB).

In addition, these pixel values are used to extract useful features using CNNs.

A. Artifical Neural Network

Artificial neural networks are connections of neurons that mimic the structure of the human brain. Each connection in a neuron sends information to another neuron. **The input** is given to the first layer of the neuron, which **will process** them and **send** them to another layer of the neuron called the hidden layer. After processing the information through several layers of the hidden layer, the information is passed to the final output layer.

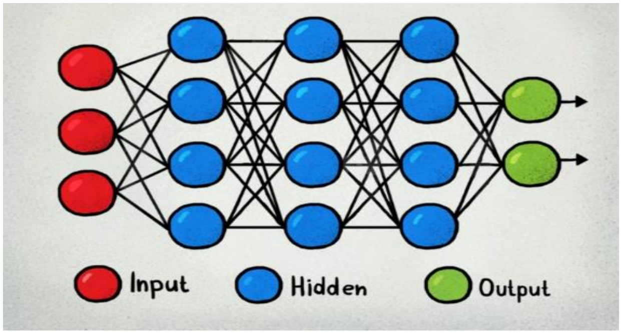

Fig.4. Artificial Neural Network

B. Convolution Neural Networks

- Unlike a **regular** neural network, neurons in the CNN layer are arranged in three dimensions: width, **height** and depth. **The neurons** in a layer are connected to **only** a small **region** of the layer (window size).
- Before that, instead all neurons **are** fully **connected. Also,** at the end of the CNN architecture, the final output layer has dimensions (number of **layers)** to reduce the **overview** to a single vector of **layer** scores.

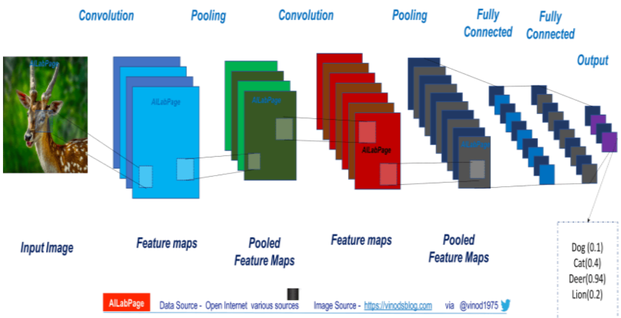

Fig.5. Convolutional Neural Network

a) Convolution Layer

The convolutional layer uses a small window size (usually 5 * 5 in length) that extends to the depth of the input matrix. This class includes window-sized learner filters. In each iteration, we move the window incrementing [usually 1] to compute the product of the filter input and the input value at a particular position. Continue this process to create a two-dimensional activation matrix that reflects the response of this matrix at each spatial location.

A filter is activated when a visual feature is displayed. REMOVE Edges in a particular direction or points of a particular color.

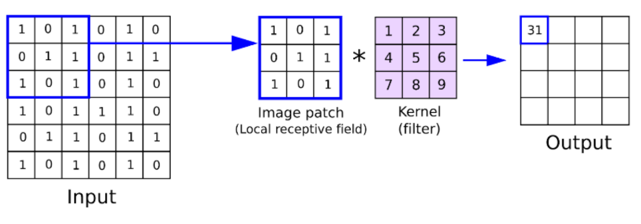

Fig.6. Convolutional Layer

b) Pooling Layer

Use a pooling layer to reduce the size of the activation matrix and possibly reduce the parameters that can be learned from it.

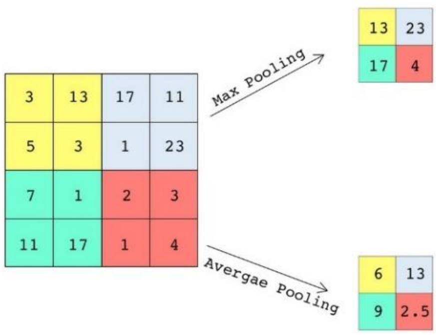

Fig.7. Maximum and Average Pooling

Maximum Aggregate uses the window size [Example: 2*2 size example window] and only takes up to values. If you close this window properly and continue this process, you will end up with an activation matrix half the size of the original size. Aggregate average takes the average of all values in the window. In the convolutional layer, neurons are only connected to the local region, but in a fully connected region, all inputs are properly connected to the neuron.

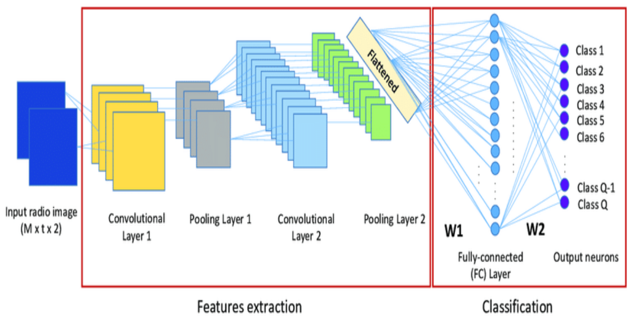

Fig.8. CNN Architecture

After getting the value of fully connected layers, connect them appropriately to the last layer of the neuron (a number equal to the total number of layers). This predicts the probability that each image belongs to a different class.

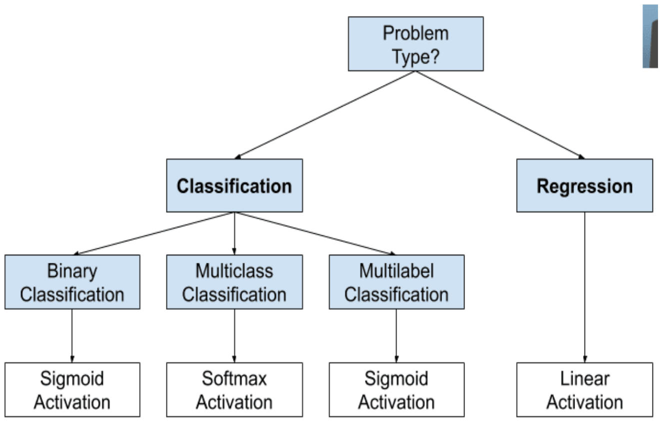

Fig.9. Activation Functions for different output layers

C. Temsor Flow

TensorFlow is an open-source digital computing software library. First identify the nodes of the computation graph, then the actual computation is done in the session. TensorFlow is widely used in machine learning.

D. Keras

Keras is a high-level neural network library written in Python and acts as a wrapper for TensorFlow. That is, to build a neural network with minimal lines of code, used when you want to build and test quickly. The class, target function, activation function, optimizer, etc., contain implementations of commonly used neural network elements. Because this tool is easy to use image and text data.

E. OpenCV

OpenCV (OpenSource Computer Vision) is an open-source library of programming functions used for real-time computer vision. It is mainly used to analyze functions such as image processing, video recording, and face and object recognition. Written in C, with main interface, Python, Java and MATLAB/OCTAVE bindings.

IV. DATASETS AND EXPERIMENTATION

A. MediaPipe Hand Gesture Recognition

The dataset contains several sequences of pre-processed Range Doppler images. Each sequence was recorded in a single data file in HDF5 format. The filename is defined as [Gesture ID] _ [Session ID] _ [instance ID] .h5. The Range Doppler image data of a particular channel can be accessed using the dataset name ch [Channel ID]. Labels are accessed by the dataset name label. The Range Doppler image data array is of the form [number of frames] * 102
(can be reshaped to a 2D Range Doppler image at 32 * 32).

Dataset session arrangement for evaluation.

- 11 (gestures) * 25 (versions) * 10 (users) for cross-evaluation: session 2 (25), 3 (25), 5 (25), 6 (25), 8 (25), 9 (25), 10 (25), 11 (25), 12 (25), 13 (25).
- 1 (gestures) * (50 (versions) * (sessions) 25 (instances) * 2 (sessions))) to cross-evaluate a user's session: session 0 (50), 1 (50), (50), 7 (50), 13 (25), 1 (25).

Each column in the dataset represents a gesture, and we take a snapshot of three important steps for each gesture. The gesture label is represented by the number in the circle above. Please note that the order of the gesture labels is different from the order on the paper, because we grouped the gestures in the paper. Sequences with Gesture ID 11 are background cues without the presence of a hand.

The Soli sensor is a solid-state millimeter wave radar. Conventional radar approaches rely on high spatial resolution to distinguish many rigid moving targets (e.g., aircraft). In contrast, Soli uses high temporal resolution priority detection to detect subtle, non-rigid movements. Soli uses a single, wide antenna beam to illuminate the entire hand as modulated pulses are transmitted at a very high repetition rate (between 110kHz).

The received raw signal, consisting of superposition of reflections from scattering centers in the radar antenna beam, is then processed into some abstract representation of the signal. The high temporal resolution allows a combination of fast and slow processing times to map the scattering central reflection into interpretable dimensions. Range Doppler images (RDI) are obtained by mapping the received energy in two-dimensional space to radial distance (or range) and velocity. Pixel intensity is reflected energy; horizontal axis is velocity; vertical axis is range.

Short-range radar data does not directly contain shape information and therefore many existing algorithms are not applicable.

The data processing steps are:

- Signal transformations (e.g., Fourier transform).
- Extraction feature.
- Frame level classification.

Each data point is a variable length sequence of 102
dimensional vectors reshaped into [32,32] 2d Doppler maps [range, velocity].

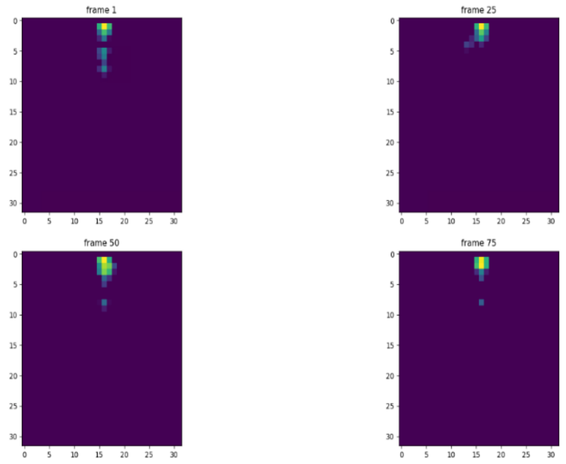

Fig.10. Radar Gesture Intensity

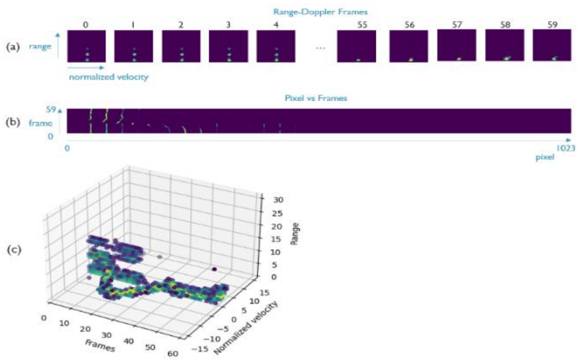

Fig.11.a Visualising Radar Data

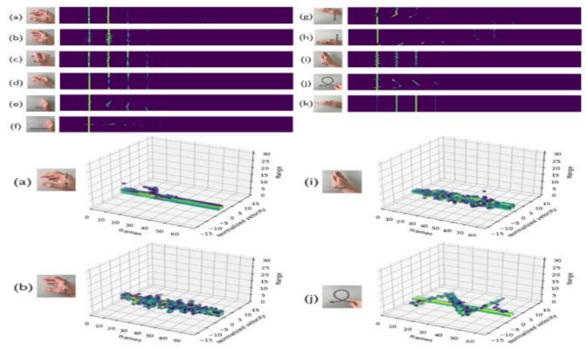

Fig.11.b Visualising Radar Data

B. Mediapipe Hand Gesture Recognition

Gesture recognition is an active research area in human-computer interaction engineering. It has many uses such as controlling virtual environments, translating sign language, controlling robots and producing music. This hand gesture recognition machine learning project uses OpenCV and Python's MediaPipe framework and TensorFlow to build real-time hand gesture recognition.

OpenCV is a real-time computer vision and image processing framework built on top of C/C. However, it is used in Python via the OpenCV Python package.

a) Mediapipe

MediaPipe is a customizable machine learning solution framework developed by Google. It is an open source and extremely lightweight cross-platform framework. MediaPipe comes with pre-trained ML solutions like face detection, pose estimation, hand detection, object detection and more.

b) Tensor Flow

TensorFlow is an open-source deep learning and machine learning library developed by the Google Brains team. It can be used for a variety of tasks, but is specifically focused on deep neural networks.

Neural network is also known as artificial neural network. It is a subset of machine learning and forms the core of deep learning algorithms. The concept of neural networks is inspired by the human brain. It mimics the way biological neurons signal each other. A neural network consists of a layer of nodes consisting of an input layer, one or more hidden layers, and an output layer.

We will first use MediaPipe to recognize the hand and **its** key points. MediaPipe returns a total of 21 key points for each detected hand.

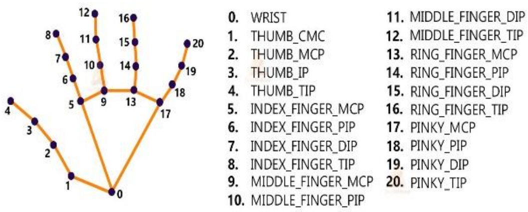

Fig.12. Finger prints detected by MediaPipe

V. Result And Analysis

A. Radar Approach

For this approach Google Deep Soli Dataset has been used. Deep Soli Radar Gesture Data have few classes, these classes were classified using the deep learning. The Deep Soli was in HDF5 format, this format is converted to make it ready to be fed into a neural network. The figure 13 depicts the code used to convert the HDF5 Google Deep Soli Dataset.

```python
# Demo code to extract data in python
import h5py

use_channel = 0
with h5py.File(file_name, 'r') as f:
    # Data and label are numpy arrays
    data = f['ch{}'.format(use_channel)][()]
    label = f['label'][()]
```

Fig.13. Code to extract data in python

The figure 1 shows the different classes of gestures available in the Google Deep Soli Gesture Dataset. Used the three main features of the radar data that are the Frames, Normalised Velocity and Range. These three features were fed to the neural network model.

A few models like CNN, 2D CNN, CNN with pooling/ without pooling and other different deep learning neural networks were trained. The 3D CNN showed up with good accuracy. The 3D-CNN model got an accuracy of 97%. The confusion matrix of the model that we did feed the features to the 3D CNN is shown in the figure 14.

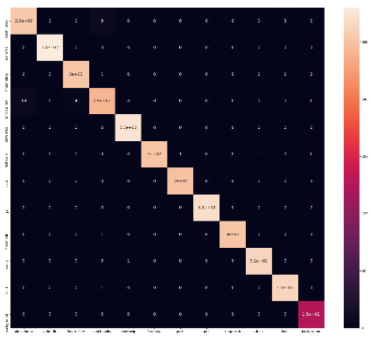

Fig.14. Confusion Matrix of 3D CNN model

The Radar Based Hand Gesture Recognition is good but when it comes to static gestures, the model failed in that. As radar has its advantages like it can survive in any condition and can penetrate through few mediums but when it comes to the static gestures it won't be able to detect the gestures. All it can detect are the dynamic gestures.

B. Camera Approach

Camera Approach was a lot more into image processing. For this approach the preferred data is American Sign Language. As we discussed it was more into finger spelling. The model is all about a bounding box. This approach was able to detect the gesture and then form a sentence and then it will dictate too. The person has to place his hand in the bounding box and then the background will be removed and then a mask is applied to the image to extract the features from the image as shown in the figure 15.

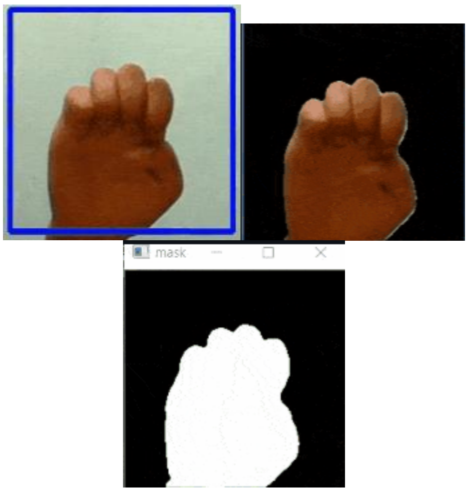

Fig.15. RGB Image, Image after removing background noice and then the Image of Alphabet E after applying mask.

Then the image goes through the CNN model and gets classified and then the formed sentence will be tranlated using the python text to speech module pytts. This model was good but the model to detect the gestures the images need to be trained. To get good predictions the model must be trained with many images and for every gesture with different possibilities a huge data must be fed to the model for training. This problem led us to the next approach.

C. MediaPipe Approach

This approach is definitely better than the other two in terms of accuracy, latency, and robustness. The Mp.solution.palms module acts as a reputation algorithm. So we create the entry and save it to mpHands.

Using the mpHands.Hands approach, we configured this instance. The first argument is max_num_hands, which means the largest range of hands that can be detected across the instance in an unmarried setting. MediaPipe can position more than one palm in an unmarried context, but most of us will only position one hand at a time in this project. Mp.solutions.drawing_utils will draw the detected key elements for us so we don't have to draw them manually.

Using the load_model function, we load the pre-trained TensorFlow model. The Gesture.names file contains the names of the gesture classes. So we first open the file using python's built-in open function and then read the file. Then we read the file using the read() function. The model can recognize 10 different gestures like: ["ok", "peace", "like", "dislike", "call me", "stop", "kick", "long live" , "holding hands", " smile "]

Then we detect the key points of the hand, then we recognize the gestures. This method really doesn't require much training and it has better real-time use cases than the radar approach and the camera approach.

Fig.16.. Gesture Recognition using MediaPipe

VI. Conclusion And Future Enhancement

AI-based hand gesture recognition through different approaches. We tried different approaches to AI-based hand gesture recognition. First, we tried radar gesture recognition with Google's Deep Soli Radar Gesture Dataset. Although we have good accuracy there, radar data does not recognize static gestures and radar has its advantage as it can survive in any climate and it can also penetrate some vehicle. Sine, radar cannot recognize static gestures and radar data requires a lot of computing power.

Now we tried the camera approach where we have to recognize American Sign Language, we form 10 digits and then we also put a translator into the program and then we also have a good accuracy there, but we have about 60% accuracy on it. Since then, it's very time-consuming because we have to train a lot of gestures and then we have to preprocess the image, the captured image then goes through a filter to remove the background and then is the mask to extract key features and then the images will go through CNN to classify the gestures there, and then the sentence can be translated text to speech.

The third approach uses MediaPipe, MediaPipe recognizes the key points of the hand and then trained models are used to detect the hand gestures there. Due to the inconveniences, we encountered when running the model, we believe that adding images of all layers with different contexts and spacing will cover a wider area and flexibility for live performances. To further improve the model, you need to add the mirror image as part of the left data set to do data expansion to make it translate invariant/equivalent. This prevents the model from overfitting certain backgrounds, spacing, and hand positions in the image. This allows you to merge the existing model with the new data set to improve the prediction accuracy of each character.

In general, the more images with different characteristics, the better the results. Instead of just recognizing letters, we can try to extend the model to recognize coherent words, phrases, and sentences. Once we have developed a model that generates the text/annotations that form logical sentences, we can continue with natural language processing so that we can run various analyzes based on the text we have extracted. output, such as sentiment analysis to find context. and the feelings behind the words. This is significant because the model will combine two different cognitive computing methods: image recognition and text analysis. In other words, this sample will become one of the representative examples of how to extract text from image.

VII. References

[1] Lan, Shengchang, Lui, Zonglong, Yao, Kai and Chen, Weichu, "Gesture Recognition Using 2 GHz Threedimensional Radar Array," in *MTTS International Microwave Symposium*, USA, 2018.

[2] Fanchen Kong, Jingcheng Deng and Zichuan Fan, "Gesture recognition system based on ultrasonic FMCW and ConvLSTM model," *Measurement* , vol. 190, no. 110743, pp. 0263-2241, 2022.

[3] H. Kulhandjian, P. Sharma, M. Kulhandjan and C. D'Amours, "Sign Language Gesture Recognition Using Doppler Radar and Deep Learning," in *Globecom Workshops (GC Wkshps)*, 2019.

[4] Cai, Xiaodong, Ma, Jingyi, Liu, Wei, Han, Hemin and Ma, Lili, "Efficient convolutional neural network for FMCW radar based hand gesture recognition," in *Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, 2019.

[5] E.D.G. Sanjeewa, K.K.L. Herath, B.G.D.A. Madhusanka, H.D.N.S. Priyankara and H.M.K.K.M.B. Herath, "Chapter 14 - Understanding the hand gesture command to visual attention model for mobile robot navigation: service robots in domestic environment," 2021, pp. 287-310.

[6] A. Helen Victoria and G. Maragatham, "Gesture recognition of radar micro doppler signatures using separable convolutional neural networks," in *Materials*, 2021.

[7] Yang, Y, Li, J and Li, B, " MDHandNet: a lightweight deep neural network for hand gesture/sign language recognition based on micro-doppler images," in *World Wide Web*, 2022.

[8] Ing Jyh Tsang, Federico Corradi, Monalis Sifalakis, Werner Van Leekwijck and Steven Latre, "Radar-Based Hand Gesture Recognition Using Spiking Neural Networks.," in *18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019.

[9] "https://www.researchgate.net/publication/354463749_Dynamic_Hand_Gesture_Recognition_Using_3D-CNN_and_LSTM_Networks/citations".

[10] Wang, Jixin, Liu, Tingting and Wang, Xuan, "Human hand gesture recognition with convolutional neural networks for K-12 double-teachers instruction mode classroom," *Infrared Physics & Technology*, pp. 103-464, 2020.

[11] S. Hazra and A. Santra, "Radar Gesture Recognition System in Presence of Interference using Self-Attention Neural Network," *18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1409-1414, 2019.

[12] R. R. Subramanian, M. S. Murali, B. Deepak, P. Deepak, H. N. Reddy and R. R. Sudharsan, "Airline Fare Prediction Using Machine Learning Algorithms," in *4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2022.

[13] R. R. Subramanian, C. S. Niharika, D. U. Rani, P. Ravani and K. P. L. Syamala, "Design and Evaluation of a Deep Learning Algorithm for Emotion Recognition," in *5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2021.

[14] R. R. Subramanian, B. R. Babu, K. Mamta and K. Manogna, "Design and Evaluation of a Hybrid Feature Descriptor based Handwritten Character Inference Technique," in *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, 2019.

[15] R. R. Subramanian, D. Achuth, P. S. Kumar, K. Naveen Kumar Reddy, S, Amara and A. S. Chowdary, "Skin cancer classification using Convolutional neural networks," in *11th International Conference on Cloud Computing, Data Science & Engineering*, 2021.

[16] R. Raja Subramanian and V. Vasudevan, "A deep genetic algorithm for human activity recognition leveraging fog computing frameworks," *Journal of Visual Communication and Image Representation*, vol. 77, pp. 1047-3203, 2021.

[17] R. Raja Subramanian and Karthick Seshadri, "Design and Analysis of a Hybrid Hierarchical Feature Tree based Authorship Inference Technique," *Advances in Data and Information Sciences, Springer*, vol. 2, pp. 89-104, 2018.

[18] R. Raja Subramanian, H. Mohan, A. Mounika Jenny, D. Sreshta, M. Lakshmi Prasanna and P. Mohan, "PSO Based Fuzzy-Genetic Optimization Technique for Face Recognition," in *11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2021.

[19] Raja Subramanian R and Vasudevan V, "An Ensemble Deep Learning Model for Activity Recognition Leveraging IoT and Fog Architectures," *In: Gunjan V.K., Zurada J.M. (eds) Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough. Studies in Computational Intelligence*, vol. 956, no. https://doi.org/10.1007/978-3-030-68291-0_11, 2021.

[20] S. Amara and R. R. Subramanian, "Collaborating personalized recommender system and content-based recommender system using TextCorpus," in *6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2020.